

Choosing reusable software strategies

Mannion, Mike; Savolainen, Juha

Published in:

SPLC '16: Proceedings of the 20th International Systems and Software Product Line Conference

DOI:

[10.1145/2934466.2934494](https://doi.org/10.1145/2934466.2934494)

Publication date:

2016

Document Version

Author accepted manuscript

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):

Mannion, M & Savolainen, J 2016, Choosing reusable software strategies. in *SPLC '16: Proceedings of the 20th International Systems and Software Product Line Conference*. Association for Computing Machinery (ACM), pp. 227-231, 20th International Systems and Software Product Line Conference, Beijing, China, 16/09/16.
<https://doi.org/10.1145/2934466.2934494>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

Choosing Reusable Software Strategies

Mike Mannion
Glasgow Caledonian University
70 Cowcaddens Road
Glasgow, UK
+44 141 331 3285
m.a.g.mannion@gcu.ac.uk

Juha Savolainen
Roche Diagnostics International Ltd
Forrenstrasse 2
6343 Rotkreuz, Switzerland
+41 79 710 82 58
Juha.Savolainen@Roche.com

ABSTRACT

For many organisations, choosing a reusable software strategy such as whether to be developing products, platforms or components, or some combination of these is not straightforward. The appropriateness of the choice can also change as an organisation's internal and external business environment context changes. In this paper we provide a management tool to help guide that decision making. We set out four broad types of business strategy and map these against four different types of reusable software development strategy. The four types of business strategy correspond to different business environments which are in turn characterised by different combinations of market predictability (low to high) and an organisation's ability to influence it (low to high). To demonstrate the framework as an analytical tool we have mapped examples of different organisations reusable software strategies and explained some circumstances in which that organisation's strategy may change.

Keywords

Reuse, software, strategy.

1. INTRODUCTION

For many organisations, choosing a reusable software strategy such as developing products, platforms or components, or some combination of these is not straightforward. The appropriateness of the choice changes as an organisation's business environment context changes i.e. the external market factors and the organisation's strengths, weaknesses and plans.

In this paper we provide a management tool to help guide reusable software strategy decision making. We set out four broad types of business strategy [1] and map these against four different types of reusable software development. The four business strategy types correspond to different business environments, which are in turn characterised by different combinations of market predictability (low to high) and an organisation's ability to influence it (low to high). Our Research Question was framed as: *what reusable software development strategy is appropriate for different business market environments?*

To demonstrate the framework as an analytical tool we have mapped examples of different organisations reusable software strategies and explained some circumstances in which that organisation's strategy may change.

2. REUSABLE SOFTWARE PRODUCT DEVELOPMENT

In ISO/IEC 26550:2015 [2] a *product* is a software system or a hardware and software system, and a product line is a set of products and/or services sharing explicitly defined and managed common and variable features and relying on the same domain architecture to meet the common and variable needs of specific markets. The domain architecture in this definition is often referred to as a *platform* which consists of one or more *components*. When an organisation embarks on a reusable

software strategy, the focus of the endeavour normally covers one or more of three categories: products, platforms or components.

For example in the 1990s and 2000s Nokia created a novel affordable mobile phone product, and as sales grew they created a wide range of mobile phone products, most of which were built on one of three platforms, Series 30, Series 40 or Series 60, which in turn made use of a common operating system component (Symbian) and a standardized GSM software stack component to allow basic calls and data transfers to take place in GSM networks. The product, the platform and the operating system component were developed by Nokia and the software stack was developed by a third party.

This example illustrates four different types of reusable software development strategy:

- *New product development*: a new product that disrupts the market and then becomes product line
- *Product line development*: a new product that is a variation on products that already exist
- *Platform-led development*: a platform architecture to support the development of existing or new products
- *Component-led development*: one or more components to support the development of products.

The two examples also show how each organisation started life making a single new product that evolved into a product line. As the market grew and they gained market share, the organisations grew larger in size, shape and capability, and were able to successfully transition into being able to trade in more than one reusable software development category. For many organisations however, making a decision about whether to be developing products, platforms or components, or some combination of these is often not straightforward. It is also a decision that needs to be revisited regularly either as part of product roadmap evolution or even simply as a reaction to one or more significant events. Whilst assessing the internal organisation context such as size, shape and capability is very important, a critical assessment of the evolving market, and the position of the organisation's offerings within it, is essential. Having some criteria to guide that market assessment can be vital.

3. BUSINESS ENVIRONMENTS AND STRATEGIES

Choosing a business strategy and a corresponding reusable software development strategy are not straightforward. It depends on several contextual factors including strategic vision, organizational capability, an assessment of the business environment, and other broader socio-economic factors. Recently Reeves et al [2] published a strategy decision toolkit that focused on assessing business environments. From a detailed survey of 150 firms and in-depth interviews with many CEOs, they found that different strategy types are being deployed for product and service offerings in accordance with these organisations'

assessments of the external business environments in which their offerings are positioned.

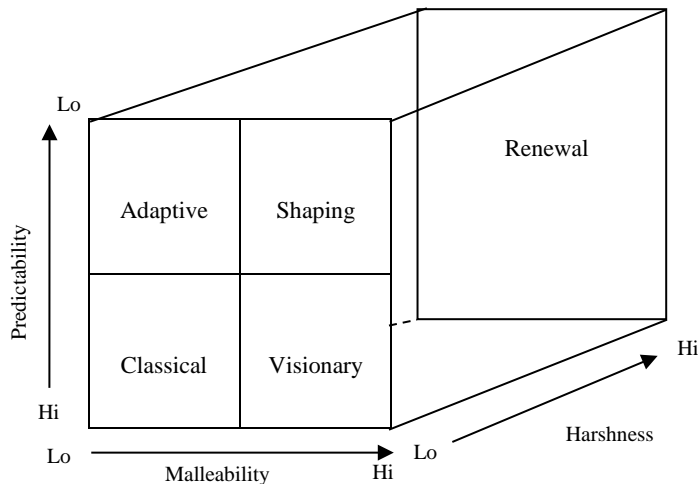


Figure 1. Business Environments and Strategies

These business environment assessments are characterised along three different dimensions: predictability, malleability and harshness. Predictability means the extent to which the market can be forecast with a reasonable degree of accuracy and hence the long-term planning is a valuable activity. Malleability means the degree to which the market can be influenced by your organisation either alone or in partnership with others particularly in the context of significant disruptions like communications technology changes, globalisation, safety and security. Harshness means the degree to which the market has suddenly changed so dramatically that your organisation can barely survive. Strategy choice should fit the business environment. Figure 1 shows these three dimensions each on a scale of low to high. When the harshness of the broader business market environment is low (i.e. conditions are favourable) then there are four different business strategies that vary depending on predictability and malleability: Visionary, Classical, Adaptive, Shaping. When the broader business environment is harsh, each of the four business strategies reverts to some form of Renewal or survival strategy.

3.1 Classical Strategy

The business environment assumptions for a Classical Strategy are that the market changes can be reasonably well anticipated (high predictability), and the basis of competition is reasonably stable (low harshness) but the ability to directly influence the market itself is constrained (low malleability).

Deploying a Classical Strategy means that competitive advantage is based on cost or differentiation, and market leadership is often gained through size or unique capabilities.

For example, in the telecommunications infrastructure industry, there are few major players because contenders need to be large enough with sufficient cash to absorb the costs of developing and evolving large-scale transmission networks. However even with few players, although it is reasonably straightforward to predict major changes in the market it is more difficult for any single organisation to influence it. Malleability increases however if the major players form a cartel.

3.2 Visionary Strategy

The business environment assumptions for a Visionary Strategy are that the market changes can be well anticipated (high

predictability), the basis of competition is reasonably stable (low harshness), and the ability to directly influence the market itself is not constrained (high malleability). However this view of high malleability often just means that the leader(s) of an organisation has sufficient self-belief in themselves and their product to shape the market even if the same market looks less malleable to others.

Deploying a Visionary strategy means disrupting the market by introducing a revolutionary new product or business model and gaining competitive advantage from being first to market either by cost or differentiation. Market leadership is developed through aggressive sales growth, sometimes initially at the expense of profit margins.

For example, Netflix created significant disruption to traditional TV broadcasting and cable operators by its Internet subscription based business model. Whilst now considered a major provider of programmes and movies, it required considerable financial support to sustain its business model in its early years.

3.3 Adaptive Strategy

The business environment assumptions for an Adaptive Strategy are that the market changes cannot be well anticipated (low predictability), and although the basis of competition is reasonably stable (low harshness), the ability to directly influence the market itself is also constrained (low malleability).

Deploying an Adaptive strategy means generating a number of options, evaluating them and selecting the ones most likely to scale up and be exploited. Competitive advantage is gained by differentiation. Market leadership is often gained by organisations that are able to complete this option generation-evaluation-selection iteration frequently and accurately. Such organisations create targeted experiments [3] that test key assumptions about the market. In a targeted experiment the intention is to acquire some evidence about a product's potential from a small presence in the market with minimal investments. This contrasts with larger-scale strategic initiatives in the Classical strategy in which the organisation believes that it can predict the market and is willing to invest heavily in a large product line to make it happen.

For example, in the fashion industry, predicting what will be a seasonal success is difficult. Zara, the flagship chain store of the world's largest apparel retailer Inditex, addressed this by having short supply chains and producing only small batches, scaling up only those that have good sales. For example much of the consulting or engineering service industry falls into this category.

3.4 Shaping Strategy

The business environment assumptions for a Shaping Strategy are that the market changes cannot be well anticipated (low predictability), and although the basis of competition is reasonably stable (low harshness), there is an opportunity to directly influence the market (high malleability), often because the cycle of the market is at an early point in its development and can be shaped.

In a Shaping strategy an organisation tries to handle the unpredictability in the market by shaping it to its advantage. Unlike in the Visionary strategy where an organisation orchestrates the change itself, in a Shaping strategy the organisation tries to orchestrate the change by creating a platform ecosystem that is beneficial for all participants.

Deploying a Shaping strategy means forming an alliance with other organisations and building an agreed development platform

through which they each can exercise complementary capabilities. Competitive advantage is gained by platform differentiation. Market leadership is gained using the combined sales and marketing power of all organisations to promote the platform.

For example, in the automotive industry, many automobile original equipment manufacturers (OEMs), whilst principally deploying Classical strategies, began sharing components in an Adaptive strategy for those components which were needed for a product to compete but offered low profit margins for the OEM. That led to deeper collaborations and now some OEMs work strategically to develop component architectures for different aspects of automotive construction (a platform-led Shaping strategy) e.g. for the chassis or vehicular software [4]. This enabled partners to benefit from reduced costs against other global suppliers not in their consortium, and to focus on competing on specific software applications in each vehicle.

3.5 Renewal Strategy

The business environment assumptions for a Renewal Strategy are that regardless of the market's predictability or the market's malleability, the basis of competition for the organisation has become unstable (high harshness) and in the near future its cost base will exceed revenues.

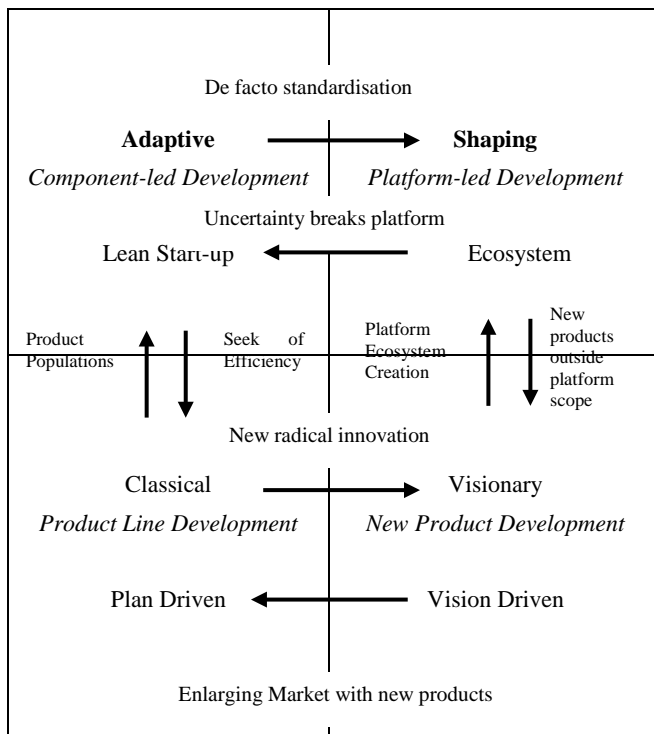


Figure 2. Reusable Software Strategies

There might be several reasons for an organisation's current predicament, including poor management, an internal or external disruptive force exposing the weaknesses of the organisation, a series of damaging external events which may be close to the business environment or maybe several steps removed from it e.g. the financial crisis of 2008 that affected all markets, or a some combination of any of these. Regardless, the organisation now needs to renew what it is doing.

For example Kodak was too slow to react to the falling costs and hence attractiveness of digital photography and its profits in

successive years slumped until they went bankrupt in 2012. The company is now out of bankruptcy and is focusing its efforts on imaging for business.

4. GUIDING REUSABLE SOFTWARE DEVELOPMENT

Figure 2 shows how we have adapted the strategy framework for guiding the appropriate deployment of a reusable software development strategy. For each of the four business strategies we have mapped a corresponding reusable software strategy. The arrows indicate factors that can drive a transition between strategy types.

4.1 Classical – Product Line Development

To execute a Classical strategy the appropriate reusable software development strategy is a systematic approach to product line engineering in which the broad processes of domain engineering and application engineering are distinguished. The intention of product line engineering is to increase efficiency by benefitting from the commonality among products. However it also requires a disciplined rigorous approach to managing variability so that different products can be created and targeted at different market segments at a lower cost per product than creating each one independently.

The high predictability of the market environment makes it possible to plan the evolution of a product line through anticipating what types of variability will be needed in both the short-term and the long-term. The low malleability of the market incentivises organisations to continue to build their own products so developers can create common assets that are shared as-is in many different products over time. The scope, scale and cost of these assets become a barrier to entry for new competitors. However, if a competitor can bring a significant disruption to the market the value of the large asset base changes overnight from a source of competitive advantage to a liability.

For example, over time Nokia created multiple product lines in the mobile phone handset market driving a highly successful classical strategy [5]. The iPhone was particularly disruptive for Nokia who were forced to re-engineer a large set of user-interface related assets to compete on the same level. Although the effort required a significant amount of time, the organisation could not just abandon its current product lines and asset bases. This led to multiple branches of assets requiring extensive parallel investments to current and future assets. Soon the organisation was drawn into a Renewal strategy.

In [6] we argued that product line businesses following a Classical strategy broadly fall into three categories and these affect how the product line is engineered. With a Cost Leadership business strategy an organization targets a market segment and competes on price by minimizing costs. The corresponding engineering strategy is characterized by making the product effective and efficient within tight budget constraints through developing a product platform in which commonality is prioritized over variability. With a Differentiation / Mid-Range / Mid-Performance business strategy an organization targets a set of market segments with a product that has a unique combination of commonality, variability and price that is valued by customers. The corresponding engineering strategy is characterized by a product platform with an architecture that has high variation management and asset-based reuse and each product is formed using pre-planned variation mechanisms and the rules governing the deployment of a common, product-line-wide architecture.

With a Differentiation / High Cost / High Performance business strategy an organization targets a market segment but provides specific products to suit the needs of individual customers generating high customer loyalty. The corresponding engineering strategy is characterized by a product platform to support a compositional approach in which variability management becomes de-centralized and the product line architecture only guarantees that the components can be connected together. Over time each product assumes its own maintenance trajectory separate from other products, there is little or no product line or reuse, and the technical strategy is effectively single-system development with some reuse.

4.2 Visionary - New Product Development

New entrants trying to break into a well-established competitive market often see a Visionary strategy as their only chance to gain significant market share, even though it often requires deep pockets to both create a unique product and persuade the market of its value. Since the goal of the new entrant is to significantly disturb the market by getting first mover advantage, the development focus is on creating a single product with little emphasis on reuse.

Apple took this strategy with its iPhone smartphone product. Apple were not in the mobile phone industry at the time but wanted to break into it as they had a vision that mobile phone and computer technologies would merge. The iPhone's novel interface, its range of applications and its sleek looks were hugely innovative. Even so it also required the passion, enthusiasm and personality of Steve Jobs to play a significant role in shaping people's opinions of the product and hence the market.

Some established organisations who are executing a Classical strategy may also develop a Visionary strategy for a new product, particularly if their product line market is reaching saturation, limiting growth, decreasing margins and increasing competition. By deploying a strategic innovation project organisations try to create a change rather than react to it [7]. This strategic innovation project is typically run independently from the rest of the organization as the intention is to disturb the current operational model. If the outcome of a strategic innovation project leads to a new product it can often change the business model of the organisation particularly if it is combined with technological innovation [8].

For example Rolls Royce transformed from a traditional airplane engine (and engine control software) sales organisation to a performance based contracting model where the revenues were tied to the amount of hours that the engine spent in the air [9]. On one hand this did not require significant investments into technical assets; beyond an ability to monitor engine health remotely, so the existing engine control software product line [10] could be used allowing fast entry to the market.

The transition from a Visionary strategy to a Classical strategy is also common. Following that initial disruption to the market, the competition will try to react to the changing business landscape by introducing their own offerings in the area. Over time market predictability and stability increase, volatility decreases and plan driven actions increase efficiency and help to achieve development objectives. While Rolls Royce has been very successful in realizing this new business model, nowadays all other aircraft engine manufacturers offer similar solutions to their customers. Thus the disruption created only a temporary competitive advantage for Rolls Royce and today aircraft engine manufacturers are again competing using Classical strategies.

4.3 Adaptive – Component-Led Development

To execute an Adaptive strategy the appropriate reusable software development strategy is a component-led development. Typically the idea is to identify and build a set of standardized, domain-specific components that are not affected by differentiating factors across product lines. These components tend to be basic services that are used by the common features of different products or even product lines. The basic services can be based on common standards that naturally make the components the same for all implementations. A good example of these types of services is the GSM software stack in mobile phones that allows basic calls and data transfers to take place in GSM networks. Different products can then build their differentiating functionality using these basic services.

In another example [11], Phillips, in the 1990s took a view that building a global product line architecture for a large set of different consumer electronics (the product population) was too complex for many reasons including hardware availability, hardware technology, market demands, organisation strategy, different parts being produced in different sub-organizations, each with their own goals, time scales, history and culture. So the organisation took an adaptive strategy and developed a common component infrastructure, Koala, in which the focus was on using components across product lines and being able to use components in a context for which they were not originally designed for.

Over time if a set of components itself is regarded as a platform, a component-led Adaptive strategy migrates towards being a platform-led Shaping strategy. Alternatively, if the market does change and starts to become more predictable, some organisations switch from an Adaptive strategy can change into Classical strategy. For example, many engineering consulting organisations have developed products tailored to specific customer needs, but while doing so have identified opportunities to create products that have a significantly broader presence in the market. For example the engineering service organisation KPIT has started to offer its own AUTOSAR middleware for car organisations [12].

4.4 Shaping – Platform-led Development

When executing a Shaping strategy the appropriate reusable software development strategy is platform-led eco-systems development.

In [13] Bosch identified some reasons why software development organisations that have built a product line development strategy might transition either instead of or in addition to a platform-led development strategy:

- the amount and cost of functionality that needs to be developed to extend the product line to satisfy the needs of an increasingly diversified customer base is more than is commercially viable
- increases the value proposition of the core platform by enabling diversity to exist and making it straightforward for customers to shape their own products can enhance market share

The platform ecosystem allows the channelling of any uncertainty in the market into innovation that keeps the ecosystem running. As the needs of customers become more complex and the need for customized solutions grows, it becomes difficult for any one organisation to fulfil all the possible needs all potential customers. Through strategic alliances about innovations a set of organisations can create value together [14]. An extension of this

idea is when organisations invite their customers to co-create innovation together [15] which further increases the value of the ecosystem.

For example when Apple created its iPhone and included several different applications, it was less clear about what subsets of applications would appeal to different market segments (low predictability) in order to grow the market. So it developed AppStore which was a platform-led Shaping strategy.

In platform based development the platform is reused in the products that are based on the platform. The value proposition for a product is that it is unnecessary to implement non-differentiating functionality. The platform ecosystem benefits from the innovations of the products developed by organisations that are true experts in their own segment. Over time shared product functionality can be implemented in the platform.

The big challenge in platform ecosystem evolution is to keep the scope of the platform well aligned with product needs. The late introduction of common product behaviour increases the effort to create new products. Too early an introduction can make the platform compete with its products. In addition, including too much functionality may lead to a bloated platform ecosystem that becomes too complex and non-competitive against other platform ecosystems.

If the platform development fails in the context of the competition an alternative is to pursue a Visionary strategy. For example a sport clock manufacturer Suunto tried to attract developers with its own Suunto “Apps” platform. However, when this approach failed they switched to a component-led Adaptive strategy through developing components for Android platforms.

4.5 Renewal Strategy

In a Renewal strategy, reusability per se is not likely to be a strong focus, just survival. Deploying a Renewal strategy means survival, cutting costs and preserving cash to be able to refocus the business towards one of the other four strategies and finding some renewed competitive advantage.

Two companies that responded poorly to the mobile web and its implications for the mobile phone handset market, Nokia and Blackberry, have embarked on renewal strategies: Nokia in telecommunications infrastructure, Blackberry mobile privacy and security features. Blackberry however intends also to remain in the handset market.

5. Conclusion

Many organisations form a view that the reuse of software will be of significant business benefit. There are different approaches to strategic reusable software development. An appropriate choice of approach will depend on the business environment context in which the organisation finds itself when making a decision. This paper offers a framework to guide that decision making.

6. REFERENCES

- [1] Reeves Your Strategy Needs A Strategy HBR Press, 2015
- [2] ISO/IEC 26550:2015 Software and systems engineering — Reference model for product line engineering and management
- [3] Sampler, J., Bringing Strategy Back, How Strategic Shock Absorbers Make Planning Relevant in a World of Constant Change, Jossey-Bass, Wiley, 2015.
- [4] www.autosar.org (last accessed 13 March 2016).
- [5] Bosch, J., Software Product Families in Nokia, Proceedings of 9th International Make Planning Conference, SPLC 2005 Rennes, France, September 26-29, 2005
- [6] Savolainen, J., Mannion, M., A Make / Buy / Reuse Feature Development Framework for Product Line Evolution”, Proceedings of 20th, International Conference on Engineering of Complex Computer Systems, 9-12 December 2015, Gold Coast, Australia
- [7] Govindarajan, V., Trimble, C., 10 Rules for Strategic Innovations, From Idea to Execution, Harward Business School Press, 2005.
- [8] Normann, R., Reframing Business, When the Map Changes the Landscape, Wiley, 2001.
- [9] Grassmann, O., Frankenberger, K., Csik, M., The Business Model Navigator, Financial Times Publishing, 2014.
- [10] Nolan, A., Abrahao, S., Dealing with Cost Estimation in Software Product Lines: Experiences and Future Directions, J. Bosch and J. Lee (Eds.): SPLC 2010, LNCS 6287, pp. 121–135, 2010.
- [11] Ommering, R.v., Building Product Populations with Software Components ICSE'02, May, 19-25, 2002, Orlando, Florida, USA
- [12] <http://www.kpit.com/engineering/automotive/autosar-in-vehicle-networks>, accessed 13.03.2016
- [13] Bosch, J Proceedings of 13th International Software Product Line Conference (SPLC 2009), August 24-28, 2009, San Francisco, CA.
- [14] Chesbrough, H., Open Services Innotation, Rethining Your Busines to Grow and Cometer in a New Era, Jossey-Bass, Wiley, 2011.
- [15] Prahalad, C.K., Ramaswamy, V., The Future of Competition, Co-Creating Univew Value with Customers, Harward Business School Publishing, 2004.